UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

BCL

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR		ATTORNEY DOCKET NO.
09/089,834	06/03/98	KNAPTON	K	INTL-0030-US

LM02/0424

TIMOTHY N TROP
TROP PRUNER HU & MILES
8550 KATY FREEWAY
SUITE 128
HOUSTON TX 77024

EXAMINER	
DAS,C	
ART UNIT	PAPER NUMBER
2762	11

DATE MAILED: 04/24/00

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
ASSISTANT SECRETARY AND COMMISSIONER OF
PATENTS AND TRADEMARKS
Washington, D.C. 20231

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Paper No. 11

Application Number: 09/089,834
Filing Date: 6/3/98
Appellant(s): Kenneth S. Knapton III

Timothy N. Trop
For Appellant

EXAMINER'S ANSWER

This is in response to appellant's brief on appeal filed on 2/7/2000.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Invention

The summary of invention contained in the brief is correct.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

The appellant's statement the grouping of claims is correct.

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) Prior Art of Record

The following is a listing of the prior art of record relied upon in the rejection of claims under appeal.

5,881,230	Christensen et al	3-1999
5,835,914	Brim	11-1998
5,682,468	Fortenbery et al	10-1997

(10) New Prior Art

No new prior art has been applied in this examiner's answer.

(11) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 1-5 and 12-17 are rejected under 35 U.S.C. 112 second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention.

With respect to claim 1, in line 4, "inserting a second object" is unclear as to where the second object is being inserted.

Claim 12 is rejected for the same reason as claim 1.

The rejection of the base claims are necessarily incorporated into their dependent claims.

2. Claim 1 is objected to because of the following informalities: It does not ended with a period. Appropriate correction is required.

3. Claims 1-6, 8, 12-13 and 15-17 are rejected under 35 U.S.C. 102(e) as being anticipated by Christensen et al, US Patent No. 5,881,230.

4. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Christensen et al, US Patent No. 5,881,230.

5. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over, Christensen et al US Patent No. 5,881,230 in view of Brim, US Patent No. 5,835,914.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(C) of this title before the invention thereof by the applicant for patent.

7. Claims 1-6, 8, 12-13 and 15-17 are rejected under 35 U.S.C. 102(e) as being anticipated by Christensen et al, US Patent No. 5,881,230.

As per claim 1, Christensen et al teach an **object oriented programming** method is shown in ABSTRACT line 1-3 ("An **object oriented programming environment** is extend to allow a client object oriented application running under a client/server operating system"), **create object and each object class having identifier** is shown in column 7 line 63-67 (" When a client application 44 asks OLE to **create an object**, OLE must first determine which server application to run. This information is stored in the operating system registry (e.g., Windows.RTM.95 registry) for each object class, and each **object class is represented by a unique GUID**") and column 7 line 36 (**"Each object class is represented by a GUID**"), **objects** are associated with a **client** is shown in column 11 line 32-40 ("The Remote Automation application ensures that objects are uniquely represented and identifiable when passed from one computer to another by assigning every object a unique GUID when it is created. GUIDs were explained in detail above. The GUIDs generated are stored in a data structure associated with the RA proxy on the **client** and in a corresponding data structure associated with all RA remote stubs to facilitate lookup by GUID while **passing remote object references**"), the GUIDs (objects) generated and associated with the client inherently including first object associated with first client and second object with second as claimed, using second object in place of the first object **without recompiling** is shown in column 2 line 64-67, column 3 line 1 and column 7 line 21-26 ("The remote automation method is used to extend the OLE object creation process and modify OLE object data. Since only OLE object data is modified, the remote automation method is compatible with existing and previously written OLE client and server object oriented applications"), object creating process inherently including creating first object and second object as claimed, modifying object data inherently including using second object in place of first as claimed and ("The Remote Automation application extends the OLE object creation process by modifying the system registry data on the client computer. Since only OLE data is modified, Remote Automation is compatible with existing OLE automation client/server applications. Existing OLE applications **do not have to be changed or recompiled** using Remote Automation").

As per claim 2, Christensen et al teach creating COM object is shown in column 5 line 61-65 ("The Component Object Model (COM) is a model used for object oriented programming. The COM specifies how objects within a single application or between applications (e.g. client/server applications) interact and communicate by defining a set of standard interfaces"). For rest of the limitations in claim 2, see the rejection of claim 1 above.

As per claim 3, Christensen et al teach **layer** class is shown in column 3 line 50-51

(" FIG. 9 is a flow chart illustrating N-tier **layering** for one embodiment of the present invention"), **globally unique identifier** is shown in column 7 line 36-37 ("Each object class is represented by a **GUID (globally unique identifier)**").

As per claim 4, Christensen et al teach layer class interfaces with one of a plurality of globally unique identifiers of objects associated with said layer class is shown in column 7 line 44-46 ("The structural definition of a GUID is manipulated by applications programs such as the Remote Automation object oriented application.) and Fig 9 part 164.

As per claim 5, Christensen et al teach using second object in place of the first object **without recompiling** is shown in column 2 line 64-67, column 3 line 1 and column 7 line 21-26 ("The remote automation method is used to extend the OLE object creation process and modify OLE object data. Since only OLE object data is modified, the remote automation method is compatible with existing and previously written OLE client and server object oriented applications"), object creating process inherently including creating first object and second object as claimed, modifying object data inherently including using second object in place of first as claimed and ("The Remote Automation application extends the OLE object creation process by modifying the system registry data on the client computer. Since only OLE data is modified, Remote Automation is compatible with existing OLE automation client/server applications. Existing OLE applications **do not have to be changed or recompiled** using Remote Automation").

As per claim 6, Christensen et al teach registering first object with a first globally unique identifier is shown in column 7 line 66-67 ("each object class is represented by a unique GUID"), each object is represented by a globally unique identifier inherently including first object registering with a first GUID and second object registering with second GUID as claimed. Selectively accessing one of said first and second objects without recompiling is shown in column 7 line 21-26 ("The Remote Automation application extends the OLE object creation process by modifying the system registry data on the client computer. Since only OLE data is modified, Remote Automation is compatible with existing OLE automation client/server applications. Existing OLE applications **do not have to be changed or recompiled** using Remote Automation").

As per claim 8, Christensen et al teach getting the identifier for each object from **database** is shown in column 7 line 27-30 ("On each host client/server operating system, preferably the operating system registry (also called the **registration database**) is used to store relevant information about object components according to their CLass IDentifier (CLSID).
Setting each GUID is said layer is shown in Fig. 9, where the application objects are in physical layers and each object has unique ID.

Claim 12 is rejected for the reasons set forth in connection of the rejection of claim 1.

As per claim 13 Christensen et al teach said objects are **COM objects** is shown in column 5 line 61-65 ("The Component Object Model (COM) is a model used for object oriented programming. The **COM specifies how objects** within a single application or between applications (e.g. client/server applications) interact and communicate by defining a set of standard interfaces").

As per claim 15, Christensen et al teach identifiers are globally unique identifier is shown in column 7 line 36 ("Each object class is represented by a GUID").

As per claim 16, Christensen et al teach one or more **instructions** as claimed is shown in column 15 line 53-55 (" A computer readable medium having **stored therein instructions** capable of causing a computer to perform the method of claim 1"), create a layer class is shown in column 3 line 50-51 (" FIG. 9 is a flow chart illustrating N-tier **layering** for one embodiment of the present invention"), globally unique identifier is shown in column 7 line 36-37 ("Each object class is represented by a GUID (globally unique identifier)").

As per claim 17 Christensen et al teach identifiers in the layer class is shown in Fig. 9, where the application objects are in physical layers and each object has unique ID.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Christensen et al, US Patent No. 5,881,230.

As per claim 7, Christensen et al do not teach creating source code version. Official notice is taken that creating source code version is well known and expected in the art. It would have been obvious to one of the ordinary skill in the art to create source code version (wrapper or interface definition) because one of the ordinary skill in the art would be motivated to provide an interface between the client and object.

10. Claim 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Christensen et al, US Patent No. 5,881,230 in view of Fortenbery, US Patent No. 5,682,468.

As per claim 9, Christensen et al teach each object has different identifier is shown in column 7 line 36 ("Each object class is represented by a GUID").

Christensen et al do not teach first object and second object and container for a software object. However, Fortenbery et al teach container for software objects, said container adapted to work with first and second object is shown in column 27 line 39-45 (" FIGS. 34a-34c illustrate the use of the IoleLocate::PointLocate function. FIG. 34a includes a first object 1590, a bolt, a second object 1600, a block with surface 1610 and 1620, and a boreline 1630. FIG. 34b includes a first object 1640 and a second object 1650. FIG. 34c includes a modified first object 1660, a bolt, and block 1600. Bolt 1590 was created in a first software application and transferred into the container, which created block 1600").

It would have been obvious to one of the ordinary skill in the art to combine Christensen's method of objects having different identifiers with Fortenbery's container for a software object because one of the ordinary skill in the art would be motivated to snap multiple objects into one given container to make the object accessed in a binary compatible fashion. With a binary compatible fashion, a new version can be plugged into an existing application that was designed and built for the old version. Thus the same identifiers and interfaces can be used for the plug-in (new version).

As per claim 10, Christensen et al teach a layer class adapted to utilize the identifier of objects is shown in column 14 line 21-27 (" Remote Automation provides the ability to quickly and easily change and configure a physical model without affecting the original logical model. Any remote object references made by a client object oriented application are uniquely represented, and known to all server object oriented applications, no matter how many physical layers are used to represent a logical model").

11. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over, Christensen et al US Patent No. 5,881,230 in view of Brim, US Patent No. 5,835,914.

As per claim 14, Christensen et al do not teach **ActiveX control**. However, Brim teaches COM objects are ActiveX controls is shown in column 6 line 54-56-58 ("An **ActiveX control** is a COM object that adheres to certain specified standards"). It would have been obvious to one of the ordinary skill in the art to combine Christensen's method of objects having different identifiers with Brim's **ActiveX control** because one of the ordinary skill in the art would be motivated to add specialized functionality in the software development tools such as animation or pop-up menus to Web pages or desktop publications.

(12) Response to Arguments

The applicant has argued:

(A) Christensen does not teach different identifiers for each of two objects.

(B) Christensen does not teach accessing one of two objects in place of the other without recompiling.

In response to appellant's arguments (A)-(B)

(A) Christensen teaches creation of different formats of object with different identifiers is shown in column 6 line 3-5 ("and allows the **creation of objects of different formats** which operate on data through defined interfaces") and each objects are uniquely represented and identifiable is shown in column 11 line 32-42 ("The Remote Automation application ensures **that objects are uniquely represented and identifiable when passed from one computer to another by assigning every object a unique GUID when it is created.** GUIDs were explained in detail above. The GUIDs generated are stored in a data structure associated with the RA proxy on the client and in a corresponding data structure associated with all RA remote stubs to facilitate lookup by GUID while passing remote object references. Thus, the client and remote server computers all understand and can identify any remote object reference by looking up its GUID in their respective data structures"), objects are uniquely represented and identifiable when passed from one computer to another by assigning every object a unique GUID when it is created, inherently including different identifiers for each of two objects. Christensen et al teach two different identifiers is shown in column 7 line 66-67 ("each object class is represented by a unique GUID"), each object is represented by a globally unique identifier inherently including first object registering with a first GUID and second object registering with second GUID as claimed.

(B) Christensen et al teach **accessing one of said first and second objects** is shown in column 5 line 5-8 ("The Windows.RTM.95 client/server operating system provides shareable resources, such as files, memory, processes and threads, **which are implemented as "objects" and are accessed by using "object services."**), accessing two objects in place of the other is shown in column 6 line 59-66 (" because the OLE proxy object 52 intercepts and forwards all calls to the real object through the OLE channel 56 to the OLE stub object 54. The OLE stub object 54 in turn calls the real object 48. The communication between the proxy and stub is managed by an OLE channel object 56, which understands how to send object information between the client 44 and server 50 applications."), OLE proxy object intercepts and forwards call to real object through the OLE channel to the OLE stub object, and stub object call the real object inherently including accessing one of two objects in place of the other, selectively accessing one of said first and second objects without recompiling is shown in column 7 line 21-26 ("The Remote Automation application extends the OLE object creation process by modifying the system registry data on the client computer. **Since only OLE data is modified**, Remote Automation is compatible with existing OLE automation client/server applications. Existing OLE applications **do not have to be changed or recompiled** using Remote Automation"). "Existing

OLE applications **do not have to be changed or recompiled** using Remote Automation" inherently including accessing one of two objects in place of the other without recompiling.

Respectfully submitted,

Chameli C. Das

10/13/2000

Conferee

James R. Trammell
Supervisory Patent Examiner
Technology Center 2700

Tariq R. Hafiz
Supervisory Patent Examiner
Technology Center 2700